UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/893,645	06/29/2001	John Charles Brook	169.2089	5805

5514 7590 06/30/2006

FITZPATRICK CELLA HARPER & SCINTO
30 ROCKEFELLER PLAZA
NEW YORK, NY 10112

EXAMINER	
SAIN, GAUTAM	
ART UNIT	**PAPER NUMBER**
2176	

DATE MAILED: 06/30/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/893,645	BROOK, JOHN CHARLES	
	Examiner	Art Unit	
	Gautam Sain	2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 June 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1- 58* is/are pending in the application.

4a) Of the above claim(s) *27,28,32,33,35-38,40-43,45,46 and 49-57* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-26,29-31,34,39, 44, 47,48 and 58* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *1/02,2/04*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. This is a Non final rejection in response to the Election on June 21, 2005.

2. Claims 1-58 are pending, of which 1-26, 29-31, 34, 39, 44, 47, 48 and 58 are elected, and claims 27, 28, 32, 33, 35-38, 40-43, 45, 46, 49-57 are withdrawn (see Election section below in paragraph 5).

3. Effective filing date is 6/30/2000.

4. Assignee is Cannon.

Election/Restrictions

5. Claims 27, 28, 32, 33, 35-38, 40-43, 45, 46, 49-57 are withdrawn from further consideration pursuant to 37 CFR 1.142(b) as being drawn to a nonelected invention, there being no allowable generic or linking claim. Election was made **without** traverse in the reply filed on Jan 19, 2006. Applicant elected Group I as including claims 1-26, 29-31, 34, 39, 47, 48 and 58. Claim 58 was mistakenly not mentioned in the requirement for Election/Restriction, however, the Applicant suggested to elect claim 58 in Group I. Accordingly, the Examiner examines claim 58 in Group I. Regarding claims 34 and 47, the Examiner initially placed claims 34 and 47 in Group II and the Applicant submitted that Claims 34 and 47 should belong in Group I and not in Group II. Accordingly, the Examiner examines claims 34 and 47 in Group I.

Claim Rejections - 35 USC § 101

6. 35 U.S.C. 101 reads as follows:

> **Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.**

6-1) Claims 1-26, 29-31 and 58 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claims 1-26, 29-31 and 58 set forth functional descriptive material but fail to set forth physical structures or materials comprising of hardware or a combination of hardware and software within the technological arts (ie., a computer) to produce a "useful, concrete and tangible" result. For example, the method of claim 1 reads on a mental construct/abstract idea or at best a computer program, per se. Claims 1-26, 29-31 and 58 are interpreted as software per se, abstract ideas or mental construct and not tangibly embodied on a computer readable medium or hardware.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

7-1) Claims 1, 2, 4-15, 18, 19, 34, 39, 44, 47, 48 and 58 are rejected under 35 U.S.C. 102(e) as being anticipated by Hind et al (US 6635088, filed Nov 20, 1998).

Regarding claim 1, Hind teaches the limitations of *parsing a markup language document comprising syntactic elements, said method comprising, for one of said syntactic elements, the steps of: identifying a type of the element; processing the element by determining a hash representation thereof if said type is a first type; and augmenting an at least partial structural representation of the document using the hash representation if said type is said first type.* For example, Hind discloses a compression of a structured document and a document type definition (Title), where XML and DTD files are compressed, reducing the file size for storage and transfer (col 3, lines 62-67). Specifically, an encoded file is read and each of the tags is located in the encoded file. Thereafter, a subprocess substitutes a unique short tag for each of the unique one of the located tags (col 4, lines 25-30). Additionally, Hind discloses a subprocess for decompressing a compressed file by reading a compressed file, locating each of the substituted short tags in the compressed file, reading the stored correspondence

between short tags and located tags, retrieving the stored tag corresponding to the located substituted short tag and substituting the retrieved tag for the located substituted short tag in the compressed file (col 4, lines 32-41). The examiner interprets Hind's short tags as equivalent to the hash representation because the short tag create a unique function for a unique string (as consistent with Applicant's specification).

Regarding claim 2, Hind teaches where parsing is event-based parsing. The examiner interprets the claimed event based parsing as parsing to provide lower-level access to XML document, facilitating parsing of documents larger than available system memory by parsing start and end of elements (see Applicant's specification, pages 1 and 2). Hind discloses opening and matching ending tags (col 2, lines 25-30) and a need to a need for reducing the length of tags for processing at a constrained-storage device (col 3, lines 1-10), where the parser reads each entity declaration in sequence (col 9, lines 5-8).

Regarding claim 4, Hind teaches wherin said first type is a declaration of said structural element. Hind discloses declaration of an entity (col 8, lines 59-65; col 9, lines 20-35 shows an example of an entity declaration).

Regarding claim 5, Hind teaches where a structural element is a tag. Hind discloses a file encoded according to a tag syntax such as XML (col 8, lines 55-59).

Regarding claim 6, Hind teaches wherein the hash representation is a unique code for said one syntactic element, said element having less than a first number of characters. Hind discloses a unique short tag for each of the located tags for which the short tag was substituted (col 4, lines 30-40) for tag attribute compression (col 4, line 43).

Regarding claim 7, Hind teaches wherein the hash representation is not a unique code for said one syntactic element, said element being constrained, to a probability level, in terms of a number of characters in the element. Hind discloses a minimum length for an entity name of one character in length, where a one character substitution string is specified in the declaration.

Regarding claim 8, Hind teaches said code comprises numeric characters. Hind discloses tags that enclose string in special characters including numbers (col 1, lines 19 –24).

Regarding claim 9, Hind teaches a sub-step of: determining an extended hash representation of both (i) said one syntactic element being a first instance of said first type, and (ii) another syntactic element being a second instance of said first type, within which said first instance, said second instance is nested. Hind discloses, in Fig 5B, a block 505' with tags "<A>" containing nested tags of type "<B>", "<C>", <D>" and <E>" which map to longer tag values seen in Fig 5A, block 505. Figures 5A and 5B show a nested structure where Customer_Nbr, Customer_Name and Ship_To_Address are nested within the block of Order.

Regarding claim 10, Hind teaches the steps of: identifying a type of the other element; and if the type of the other element is equivalent to said first type: (i) processing the other element to thereby determine a second hash representation thereof; and (ii) augmenting said at least partial structural representation of the document using the second hash representation, wherein: said processing and said second processing ensure that if a first relationship exists between the one element and the other element,

then a second relationship which is representative of the first relationship, exists between the hash representation of the one element and the hash representation of the other element. Hind discloses a method of compressing and decompressing encoded files with tags. The compressing process reads an encoded file, locates each of a plurality of tags in the encoded file, substitutes a unique short tag for each located tag in the encoded file and stores a correspondence between each of the short tag and the located tag for which it was substituted (col 4, lines 25-32). Each XML tag has an opening tag and a matching closing tag (col 2, lines 26-28). The decompressing process reads the compressed file, locates each of the substituted short tags in the compressed file, reads the stored correspondence between short tags and located tags, retrieving the stored located tag corresponding to the located substituted short tag and substitutes the retrieved tag for the located substituted short tag in the compressed file (col 4, lines 30-41).

Regarding claim 11, Hind teaches where one element is a start tag; the other element is an end tag; the hash representation of the one element is a corresponding hashed start tag, and; the second hash representation of the other element is a corresponding hashed end tag. Hind discloses opening tag and a matching closing (or "end") tag (col 2, lines 26-27). Fig 5A, items 560 and 561, show opening and closing tags, respectively. Fig 5B, items 560' and 561', show opening and closing tags after compression (col 13, lines 9 –15).

Regarding claim 12, Hind teaches where the end tag is a first modification of the start tag; and the hashed end tag is a second modification of the hashed start tag, said

second modification being representative of the first modification. Hind discloses opening tag and a matching closing (or "end") tag (col 2, lines 26-27). Fig 5A, items 560 and 561, show opening and closing tags, respectively. Fig 5B, items 560' and 561', show opening and closing tags after compression (col 13, lines 9 –15). The closing tags include a "/" symbol with an open bracket angle that designates that it is the end tag (col 2, lines 30-35).

Regarding claim 13, Hind teaches where the end tag is the same as the start tag apart from having a distinguishing character incorporated therein; and the hashed end tag is the same as the hashed start tag apart from having a distinguishing character incorporated therein. Hind discloses opening tag and a matching closing (or "end") tag (col 2, lines 26-27). Fig 5A, items 560 and 561, show opening and closing tags, respectively. Fig 5B, items 560' and 561', show opening and closing tags after compression (col 13, lines 9 –15). The closing tags include a "/" symbol with an open bracket angle that designates that it is the end tag (col 2, lines 30-35). Fig 5A shows an example where the start tag is "<order>" and the matching end tag is </order>", which is compressed in Fig 5B as "<A>" and the matching end tag is "</A>".

Regarding claim 14, Hind teaches wherin the one and the other element comprise respectively a start tag and an end tag, being a first pair of tags; corresponding hashed start and end tags for said first pair of tags are incorporated into the partial structural representation of said document; the document further includes a second pair of tags comprising a respective start tag and end tag, said second pair of tags being nested within said first pair of tags in said document, said method comprising further steps of:

processing said second pair of tags to form corresponding second hashed start and end tags; augmenting said at least partial structural representation of the document using said corresponding second hashed start and end tags so that said second hashed start and end tags indicate a nesting in relation to said hashed start and end tags for the first pair of tags which is equivalent to the nesting of said second pair of tags within said first pair of tags. Hind discloses, in Fig 5A a nested structure 505 prior to compression containing tags and in Fig 5B, the compressed file in block 505' with tags that correspond to the file shown in Fig 5A (also see col 13, lines 1-18). Specifically, Fig 5B, block 505' shows the compressed tag "<A>" containing compressed nested tags of type "<B>", "<C>", <D>" and <E>" which map to longer tag values seen in Fig 5A, block 505. Where compressed <A> corresponds to longer tag <Order>. And further, Figures 5A and 5B show a nested structure where Customer_Nbr, Customer_Name and Ship_To_Address are nested within the block of Order.

Regarding claim 15, Hind discloses a further step of concatenating the first hashed start tag with the second hashed start tag, and concatenating the first hashed end tag with the second hashed end tag, to thereby form respective extended hashed start and end tags for said second pair, wherein: said augmenting step is performed using said respective extended hashed start and end tags for said second pair, and; said extended hashed start and end tags indicate a nesting in relation to said hashed start and end tags for the first pair of tags which is equivalent to the nesting of said second pair of tags within said first pair of tags. Hind discloses, in Fig 5A a nested structure 505 prior to compression containing tags and in Fig 5B, the compressed file in block 505' with

tags that correspond to the file shown in Fig 5A (also see col 13, lines 1-18). Specifically, Fig 5B, block 505' shows the compressed tag "<A>" containing compressed nested tags of type "<B>", "<C>", <D>" and <E>" which map to longer tag values seen in Fig 5A, block 505. Where compressed <A> corresponds to longer tag <Order>. And further, Figures 5A and 5B show a nested structure where Customer_Nbr, Customer_Name and Ship_To_Address are nested within the block of Order. Additionally, in Fig 5A, block 510 is concatenated to block 505 in the longer uncompressed file, and corresponds to the compressed file in Fig 5B, concatenated block 510' and block 505'.

Regarding claim 18, Hind teaches a declaration of said structural element. Hind discloses declaration of an entity (col 8, lines 59-65; col 9, lines 20-35 shows an example of an entity declaration).

Regarding claim 19, Hind teaches wherein said structural element is a tag. Hind discloses tag syntax for the entity declaration (col 8, lines 57-62).

Regarding independent claim 34, Hind teaches a*n apparatus for parsing a markup language document comprising syntactic elements, said apparatus comprising: identifying means for identifying a type of the element; processing means for processing the element by determining a hash representation thereof if said type is a first type; and augmenting means for augmenting an at least partial structural representation of the document using the hash representation if said type is said first type.*

For example, Hind discloses a compression of a structured document and a document type definition (Title), where XML and DTD files are compressed, reducing the file size

for storage and transfer (col 3, lines 62-67). Specifically, an encoded file is read and each of the tags is located in the encoded file. Thereafter, a subprocess substitutes a unique short tag for each of the unique one of the located tags (col 4, lines 25-30). Additionally, Hind discloses a subprocess for decompressing a compressed file by reading a compressed file, locating each of the substituted short tags in the compressed file, reading the stored correspondence between short tags and located tags, retrieving the stored tag corresponding to the located substituted short tag and substituting the retrieved tag for the located substituted short tag in the compressed file (col 4, lines 32-41). The examiner interprets Hind's short tags as equivalent to the hash representation because the short tag create a unique function for a unique string (as consistent with Applicant's specification).

Hind discloses programming code embodied in a memory and accessed by the microprocessor on a network (col 7, lines 5-10) implemented as one or more computer software programs (col 7, lines 39).

Regarding independent claim 39, Hind discloses *a computer program which is configured to make a computer execute a procedure to parse a markup language document comprising syntactic elements, said program comprising: code for identifying a type of an element; code for processing the element by determining a hash representation thereof if said type is a first type; and code for augmenting an at least partial structural representation of the document using the hash representation if said type is said first type.*

For example, Hind discloses a compression of a structured document and a document type definition (Title), where XML and DTD files are compressed, reducing the file size for storage and transfer (col 3, lines 62-67). Specifically, an encoded file is read and each of the tags is located in the encoded file. Thereafter, a subprocess substitutes a unique short tag for each of the unique one of the located tags (col 4, lines 25-30). Additionally, Hind discloses a subprocess for decompressing a compressed file by reading a compressed file, locating each of the substituted short tags in the compressed file, reading the stored correspondence between short tags and located tags, retrieving the stored tag corresponding to the located substituted short tag and substituting the retrieved tag for the located substituted short tag in the compressed file (col 4, lines 32-41). The examiner interprets Hind's short tags as equivalent to the hash representation because the short tag create a unique function for a unique string (as consistent with Applicant's specification).

Hind discloses programming code embodied in a memory and accessed by the microprocessor on a network (col 7, lines 5-10) implemented as one or more computer software programs (col 7, lines 39).

Regarding independent claim 44, Hind teaches a *computer program product including a computer readable medium having recorded thereon a computer program which is configured to make a computer execute a procedure to parse a markup language document, said program comprising: code for identifying a type of the element; code for processing the element by determining a hash representation thereof if said type is a*

first type; and code for augmenting an at least partial structural representation of the document using the hash representation if said type is said first type.

For example, Hind discloses a compression of a structured document and a document type definition (Title), where XML and DTD files are compressed, reducing the file size for storage and transfer (col 3, lines 62-67). Specifically, an encoded file is read and each of the tags is located in the encoded file. Thereafter, a subprocess substitutes a unique short tag for each of the unique one of the located tags (col 4, lines 25-30). Additionally, Hind discloses a subprocess for decompressing a compressed file by reading a compressed file, locating each of the substituted short tags in the compressed file, reading the stored correspondence between short tags and located tags, retrieving the stored tag corresponding to the located substituted short tag and substituting the retrieved tag for the located substituted short tag in the compressed file (col 4, lines 32-41). The examiner interprets Hind's short tags as equivalent to the hash representation because the short tag create a unique function for a unique string (as consistent with Applicant's specification).

Hind discloses programming code embodied in a memory and accessed by the microprocessor on a network (col 7, lines 5-10) implemented as one or more computer software programs (col 7, lines 39).

Regarding independent claim 47, Hind teaches *An at least partial structural representation a markup language document comprising syntactic elements, said at least partial representation having been produced by a method comprising, for one of said syntactic elements, the steps of: identifying a type of the element; processing the*

element by determining a hash representation thereof if said type is a first type; and augmenting an at least partial structural representation of the document using the hash representation if said type is said first type.

For example, Hind discloses a compression of a structured document and a document type definition (Title), where XML and DTD files are compressed, reducing the file size for storage and transfer (col 3, lines 62-67). Specifically, an encoded file is read and each of the tags is located in the encoded file. Thereafter, a subprocess substitutes a unique short tag for each of the unique one of the located tags (col 4, lines 25-30). Additionally, Hind discloses a subprocess for decompressing a compressed file by reading a compressed file, locating each of the substituted short tags in the compressed file, reading the stored correspondence between short tags and located tags, retrieving the stored tag corresponding to the located substituted short tag and substituting the retrieved tag for the located substituted short tag in the compressed file (col 4, lines 32-41). The examiner interprets Hind's short tags as equivalent to the hash representation because the short tag create a unique function for a unique string (as consistent with Applicant's specification).

Hind discloses programming code embodied in a memory and accessed by the microprocessor on a network (col 7, lines 5-10) implemented as one or more computer software programs (col 7, lines 39).

Regarding independent claim 48, Hind teaches *An apparatus for parsing a markup language document comprising syntactic elements, said apparatus comprising: a processor; a memory for storing (i) the document, and (ii) a program which is configured*

to make the processor execute a procedure to parse the document; said program comprising: (i) code for identifying a type of an element; (ii) code for processing the element by determining a hash representation thereof if said type is a first type; and (iii) code for augmenting an at least partial structural representation of the document using the hash representation if said type is said first type.

For example, Hind discloses a compression of a structured document and a document type definition (Title), where XML and DTD files are compressed, reducing the file size for storage and transfer (col 3, lines 62-67). Specifically, an encoded file is read and each of the tags is located in the encoded file. Thereafter, a subprocess substitutes a unique short tag for each of the unique one of the located tags (col 4, lines 25-30). Additionally, Hind discloses a subprocess for decompressing a compressed file by reading a compressed file, locating each of the substituted short tags in the compressed file, reading the stored correspondence between short tags and located tags, retrieving the stored tag corresponding to the located substituted short tag and substituting the retrieved tag for the located substituted short tag in the compressed file (col 4, lines 32-41). The examiner interprets Hind's short tags as equivalent to the hash representation because the short tag create a unique function for a unique string (as consistent with Applicant's specification).

Hind discloses programming code embodied in a memory and accessed by the microprocessor on a network (col 7, lines 5-10) implemented as one or more computer software programs (col 7, lines 39).

Regarding claim 58, Hind teaches wherein said code comprises numeric characters. Hind discloses tags that enclose string in special characters including numbers (col 1, lines 19 –24).

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8-1) Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Hind (as cited above), in view of Fernandez et al (US 6785673, Feb 9, 2000, provisional application No. 60/181,400).

Regarding claim 3, Hind teaches a hash algorithm. Hind discloses logic for processing of substitution of references between tags (col 10, lines 27-40). Examiner interprets logic as equivalent to an algorithm and the process of substituting as equivalent to hash. Hind does not teach, but Fernandez teaches a reference to said hash algorithm dependent upon an associated Universal Reference Indicator; a reference to said hash algorithm dependent upon an associated namespace; and a reference to said hash algorithm dependent upon an associated Extended Markup Language declaration. For example, Fernandez discloses an optimization algorithm in a large scale XML publishing scenario using a hash table to assemble XML objects which require the XML view to fit in main memory (col 33, lines 9 – 36) utilizing a URI, a namespace and a

declaration "http://acme.com:/products.xml" (col 11, line 50). The examiner interprets this http address as containing a URI "acme.com", a namespace of "acme" and the entire line 50 as the declaration.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hind to include a URI, a namespace and a declaration as suggested by Fernandez, providing the benefit of utilizing hash tables for allowing XML views to fit into a main memory (col 33, lines 30-35).

8-2) Claims 16, 17, 20-26 and 29-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hind (as cited above), in view of Applicant Admitted Prior Art (see specification Fig 2(a) and 2(b) and pages 1-3 and 18-23)(hereinafter "AAPA").

Regarding claim 16, Hind does not expressly teach, but AAPA teaches wherein the augmenting step is succeeded by a well-formedness checking step against a syntactic rule, said well-formedness checking step comprising checking said at least partial structural representation of the markup language document against the syntactic rule by numerically comparing corresponding hashed representations of elements in said at least partial structural representation of the markup language document. AAPA discloses in Fig 2(a) and 2(b) a prior art SAX parser which supports well-formedness and or validation checking sub-processes (see specification, page 18, lines 18-19) to ensure that the document meets appropriate "well-formedness constraints" as defined by XML 1.0, to test the document for compliance with general structure rules, particularly whether tags in a document have been properly nested (see specification, page 19, lines 11-20). The Examiner notes that the Applicant expressly notes Fig 2(a)

and 2(b) as prior art as well as the description in the specification on page 18 onwards in reference to Fig 2(a) and 2(b).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hind to include well-formedness as taught by AAPA, providing the benefit of ensuring that the document meets appropriate constraints (AAPA, specification, page 19, line 13).

Regarding claim 17, Hind teaches wherein said numerically comparing step is succeeded by a further step of string-comparing, in accordance with said syntactic rule, corresponding non-hashed representations of elements not of said first type. Hind discloses entries in a table of strings where once all the string have been located and the strings that meet the condition have been entered into the string table, the logic processes this string table, using it to perform the substitution of entries for strings within the input file (col 10, line 66 – col 11, line 15)

Regarding claim 20, Hind does not expressly teach, but AAPA suggests a method, comprising a further step of: checking the well-formedness of said at least partial structural representation of the document against a syntactic rule. AAPA discloses in Fig 2(a) and 2(b) a prior art SAX parser which supports well-formedness and or validation checking sub-processes (see specification, page 18, lines 18-19) to ensure that the document meets appropriate "well-formedness constraints" as defined by XML 1.0, to test the document for compliance with general structure rules, particularly whether tags in a document have been properly nested (see specification, page 19, lines 11-20). A validation check involves a comparison of syntactic elements in a

document against validity constraints defined in a Validation Reference Document ("VRD") such as a DTD.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hind to include well-formedness as taught by AAPA, providing the benefit of ensuring that the document meets appropriate constraints (AAPA, specification, page 19, line 13) for the DTD taught by Hind (see Hind, col 3, lines 65-67).

Regarding claim 21, Hind does not expressly teach, but AAPA suggests wherein the syntactic rule relates to proper nesting of tags and said checking step comprises sub-steps of: performing a numerical comparison across hashed tags in said at least partial structural representation of the document to thereby identify said first hashed start and end tags and said second hashed start and end tags; and verifying that the second hashed start and end tags indicate a proper nesting in relation to said first hashed start and end tags. AAPA discloses in Fig 2(a) and 2(b) a prior art SAX parser which supports well-formedness and or validation checking sub-processes (see specification, page 18, lines 18-19) to ensure that the document meets appropriate "well-formedness constraints" as defined by XML 1.0, to test the document for compliance with general structure rules, particularly whether tags in a document have been properly nested (see specification, page 19, lines 11-20).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hind to include well-formedness as taught by AAPA, providing the benefit of ensuring that the document meets appropriate constraints for being properly nested (AAPA, specification, page 19, line 13) for the compressed and uncompressed DTD

taught by Hind (see Hind, col 3, lines 65-67).

Regarding claim 22, Hind teaches in the numerical comparison is followed by a further step of performing a string comparison, in accordance with said syntactic rule, across non-hashed parts of respective tags in said at least partial structural representation of the document. Hind discloses entries in a table of strings where once all the string have been located and the strings that meet the condition have been entered into the string table, the logic processes this string table, using it to perform the substitution of entries for strings within the input file (col 10, line 66 – col 11, line 15)

Regarding claim 23, Hind does not expressly teach, but AAPA suggests a method, comprising a further step of: checking the well-formedness of said at least partial structural representation of the document against a syntactic rule. AAPA discloses in Fig 2(a) and 2(b) a prior art SAX parser which supports well-formedness and or validation checking sub-processes (see specification, page 18, lines 18-19) to ensure that the document meets appropriate "well-formedness constraints" as defined by XML 1.0, to test the document for compliance with general structure rules, particularly whether tags in a document have been properly nested (see specification, page 19, lines 11-20). A validation check involves a comparison of syntactic elements in a document against validity constraints defined in a Validation Reference Document ("VRD") such as a DTD.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hind to include well-formedness as taught by AAPA, providing the benefit of ensuring that the document meets appropriate constraints (AAPA,

specification, page 19, line 13) for the DTD taught by Hind (see Hind, col 3, lines 65-67).

24. A method according to claim 23, wherein the syntactic rule relates to proper nesting of tags and said checking step comprises sub-steps of: performing a numerical comparison across hashed tags in said at least partial structural representation of the document to thereby identify said first hashed start and end tags and said extended hashed start and end tags; and verifying that the extended hashed start and end tags indicate a proper nesting in relation to said first hashed start and end tags.

Regarding claim 24, Hind does not expressly teach, but AAPA suggests the syntactic rule relates to proper nesting of tags and said checking step comprises sub-steps of: performing a numerical comparison across hashed tags in said at least partial structural representation of the document to thereby identify said first hashed start and end tags and said extended hashed start and end tags; and verifying that the extended hashed start and end tags indicate a proper nesting in relation to said first hashed start and end tags. AAPA discloses in Fig 2(a) and 2(b) a prior art SAX parser which supports well-formedness and or validation checking sub-processes (see specification, page 18, lines 18-19) to ensure that the document meets appropriate "well-formedness constraints" as defined by XML 1.0, to test the document for compliance with general structure rules, particularly whether tags in a document have been properly nested (see specification, page 19, lines 11-20).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hind to include well-formedness as taught by AAPA, providing the benefit of ensuring that the document meets appropriate constraints for being properly nested

(AAPA, specification, page 19, line 13) for the compressed and uncompressed DTD taught by Hind (see Hind, col 3, lines 65-67).

Regarding claim 25, Hind teaches in the numerical comparison is followed by a further step of: performing a string comparison across non-hashed parts of respective tags in said at least partial structural representation of the document. Hind discloses entries in a table of strings where once all the string have been located and the strings that meet the condition have been entered into the string table, the logic processes this string table, using it to perform the substitution of entries for strings within the input file (col 10, line 66 – col 11, line 15)

Regarding claim 26, Hind does not expressly teach, but AAPA suggests, wherein the well-formedness checking step is (b) included in a validation step against a validation reference document VRD, said validation step comprising sub-steps of: (a) processing the VRD, said processing comprising, for a syntactic element in the VRD, sub-sub-steps of: (i) identifying a type of said syntactic element of the VRD; and (ii) processing the syntactic element by determining a hash representation thereof if said type is said first type; and (b) checking said at least partial structural representation of the markup language document against the processed VRD, said checking comprising a sub-sub-step of numerically comparing corresponding hashed representations of elements.

AAPA discloses in Fig 2(a) and 2(b) a prior art SAX parser which supports well-formedness and or validation checking sub-processes (see specification, page 18, lines 18-19) to ensure that the document meets appropriate "well-formedness constraints" as defined by XML 1.0, to test the document for compliance with general structure rules,

particularly whether tags in a document have been properly nested (see specification, page 19, lines 11-20). A validation check involves a comparison of syntactic elements in a document against validity constraints defined in a Validation Reference Document ("VRD") such as a DTD. The Examiner asserts that the compressed XML file of Hind is still an XML file and subject to the same validation rules as an uncompressed XML file.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hind to include well-formedness as taught by AAPA, providing the benefit of ensuring that the document meets appropriate constraints (AAPA, specification, page 19, line 13) for the DTD taught by Hind (see Hind, col 3, lines 65-67).

Regarding claim 29, Hind teaches wherein said numerically comparing step is succeeded by a further step of string-comparing corresponding non-hashed representations of elements not of said first type. Hind discloses entries in a table of strings where once all the string have been located and the strings that meet the condition have been entered into the string table, the logic processes this string table, using it to perform the substitution of entries for strings within the input file (col 10, line 66 – col 11, line 15)

Regarding claim 30, Hind teaches a declaration of said structural element. Hind discloses declaration of an entity (col 8, lines 59-65; col 9, lines 20-35 shows an example of an entity declaration).

Regarding claim 31, Hind teaches wherein said structural element is a tag. Hind discloses tag syntax for the entity declaration (col 8, lines 57-62).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Gautam Sain whose telephone number is 571-272-4096. The examiner can normally be reached on M-F 9-5 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather Herndon can be reached on 571-272-4136. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

GS 4/22/06
gs

Heather Herndon
HEATHER R. HERNDON
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100